REMARKS

Claims 1-9, 10-14 are pending in this application. Claim 10 has been cancelled, without prejudice. Claims 1, 4, 11 and 14 have been amended to further define the invention. Support of these amendment can be found, for example, at Fig. 1, Fig. 2 and page 10, lines 4-6 of the specification. No new matter has been added. These claim amendments and cancellation are made solely to expedite the prosecution. Reconsideration of the pending claims in view of the amendments and the following remarks is respectfully requested. The various issues raised by the Office Action are addressed next in the order in which they appear in the Office Action.

112 Rejection

The Office Action rejected claim 10 under 35 U.S.C. §112, second paragraph, as being indefinite in that it fails to point out what is included or excluded by the claim language. This rejection is moot due to the abandonment of claim 10. Applicants respectfully submit that the claim 10 is abandoned, without prejudice, solely to expedite the prosecution. Applicants reserve the right to include claim 10 in a continuation or divisional application.

102 Rejections

The Office Action rejected claims 1 and 10-14 under 35 U.S.C. 102 (b) as being anticipated by Brotsky (US Patent Number 5,490,246) ("Brotsky"). Applicants respectfully submit that claims 1 and 10-14 are not anticipated by Brotsky for at least the following reasons.

First, Brotsky is directed to a graphical editor where the user creates a directed graph (the ACG) in order to describe how an image is constructed out of one or more figural elements such as graphical images. See col. 5, ln. 4-20. Brotsky's ACG only describes the input-output relationship between image data. Brotsky does not teach, disclose, or suggest analyzing software entities comprising software program code. To further explain the difference between software program code and graphical image.

- 5 -

Applicants refer to exhibit A to show an example of software program code written in "Java" (a programming language). As an person of ordinary skill in the art would understand, a piece of software program code includes instructions written in a software programming language. One piece of software code may depends on another software code by inclusion, association, or inheritance. Graphical images, on the other hand, are pictures and are NOT software program code that could be compiled and executed by a computer.

By contrast, pending claim 1 expressly requires "a <u>software analysis tool</u> comprising: means for converting software entities and their relationships into a graph having a structure of nodes interconnected by edges... <u>wherein the software entities comprise software program code</u>". Pending claim 11 expressly require "A dependency analysis system recorded on a computer-readable medium,...<u>wherein said analyzed system comprises software program code.</u>"

Accordingly, it is respectfully submitted that Brotsky does not anticipate pending claims 1 and 11 because Brotsky does not teach, disclose, or suggest analyzing the dependency of software entities comprising software program code.

Pending claim 12 depends from claim 11. Therefore, it is respectfully submitted that claim 12 is not anticipated by Brotsky for at least the reason associated with claim 11.

With respect to claim 13, it is respectfully submitted that Brotsky does not teach, disclose, or suggest a graph model data structure <u>for storing dependency information derived through the abstraction layer from third-party tools,</u> In fact, the third party software (i.e. QuickDraw) mentioned by the Office Action is an imaging software that helps a user to create images on a computer screen and is NOT a dependency analysis software. (See Exhibit B for a description of the functionalities of QuickDraw.) Thus, no dependency information can be derived from imaging software like QuickDraw.

With respect to claim 14, it is respectfully submitted that Brotsky does not teach, disclose, or suggest replacing a displayed node with one or more <u>embedded</u> child nodes in response to the user action. In this invention, replacing a node with its child node involves extracting the child graph embedded in the parent node based on the dependency

information. Brotsky only discloses <u>adding or deleting</u> nodes in response to a user action. Moreover, Brotsky's ACG is a "flat" graph in that it does not contain any node that further contains a not-displayed child ACG graph and thus it does not teach replacing a node in the ACG with <u>an embedded</u> child graph contained in the node.

Accordingly, it is respectfully requested that all of the Section 102 rejections be withdrawn.

103 rejections

The Office Action rejected claims 2-3 under 35 U.S.C. 103 (a) as being unpatentable over Brotsky in view of Guy E. Blelloch, "Provably Efficient Scheduling for Languages with fine-Grained Parallelism" ("Blelloch"). It is respectfully submitted that claims 2-3 are patentable over Brotsky in view of Blelloch for at least the following reasons.

It is first respectfully submitted that Brotsky and Blelloch are not analogous art and thus can not be combined in a 103 rejection. Brotsky, as described above, teaches a graphic editing tool. Blelloch, on the other hand, teaches a tool for dynamically scheduling the execution of tasks performed by parallel algorithms during program execution. It is obvious that a graph can not be executed and is not analogous to a task performed by parallel algorithms. Applicants therefore respectfully submit that the teachings of Brotsky and Blelloch serve completely different purpose and thus can not be combined in a 103 rejection.

Moreover, the scheduler taught by Blelloch is a "dynamic" run-time scheduler in that it allocates computing resources in a step by step fashion as the structure of the algorithm is revealed during its execution. See, e.g., Abstract on pg. 281-282. In contrast, the software analysis tool defined by claim 1 is a "static" tool in that it converts software entities including software program code and their relationships into graphs without executing those software entities.

- 7 -

Furthermore, it is respectfully submitted that Blelloch does not teach, disclose, or even suggest bi-directionally folding and unfolding a graph between meta and child levels, as required by claims 2-3. Specifically, the disclosure at pg. 301, section 5.4.2 of Blelloch only teaches "dynamically unfolding DAGs" (Directed Acyclic Graphs) whose nodes and edges are dynamically created as the program is executed. In contrast, the term "unfolding" in claims 2-3 refers to the removal of a meta node on a graph and replacing it with its embedded child graph. Therefore, "unfolding DAGs" as taught by Blelloch is entirely different from "unfolding a meta node" as taught by claims 2-3.

The Office Action rejected claims 4-9 under 35 U.S.C. 103 (a) as being unpatentable over Brotsky in view of Blelloch, further in view of Perttunen (US Patent Number 6,359,635) ("Perttunen"). It is respectfully submitted that these claims are patentable over Brotsky, Blelloch and Perttunen for at least the following reasons.

It is first respectfully submitted that Brotsky, Blelloch and Perttunen are not analogous art and thus can not be combined in a 103 rejection. Brotsky teaches a graphic editing tool. Blelloch teaches a tool for dynamically scheduling the execution of tasks performed by parallel algorithms during program execution. Perttunen teaches the presentation of categorized information, such as organization charts, in the form of graphs and charts. See, e.g., col. 1, lns. 7-9. Applicants respectfully submit that the structure and functionality of three references are distinctively different from one another and thus can not be combined in a 103 rejection.

Next, it is respectfully submitted that like Brotsky and Blelloch, Perttunen does not teach, disclose, or suggest analyzing the dependency of software entities comprising software program code. Therefore, the combined teachings of Brotsky, Blelloch and Perttunen do not render claims 4-9 unpatentable since claims 4-9 ultimately depend from patentable independent claim 1 and claim 1 expressly requires analyzing the dependency of software entities comprising software program code.

Accordingly, it is respectfully requested that all of the Section 103 rejections be withdrawn.
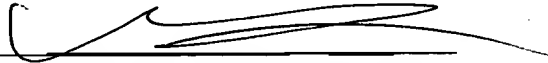
CONCLUSION

In light of the above, it is respectfully submitted that the present application is in condition for allowance. Favorable disposition is respectfully requested. Should the Examiner have any questions or comments concerning this submission, or any aspect of the application, the Examiner is respectfully invited to call the undersigned at the phone number listed below.

No fee other than the time extension fee is believed due at this time. Should any fees be required, please charge such fees to Morgan, Lewis & Bockius LLP Account No. 50-0310.

Respectfully submitted,

Dated:_____April 29, 2004_____

(Kevin) Xin He   (Reg. No.51,794)

Morgan, Lewis & Bockius LLP
101 Park Avenue
New York, New York 10178-0060
(212) 309-6623

1-NY/1774044.1